



UNITED STATES PATENT AND TRADEMARK OFFICE

A

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/422,998	10/21/1999	DANIEL W. HEPNER	10990763-1	6218

7590 10/31/2005

HEWLETT PACKARD COMPANY
INTELLECTUAL PROPERTY ADMINISTRATION
3404 E HARMONY ROAD
P.O. BOX 272400
FORT COLLINS, CO 80528-9599

EXAMINER

PHAM, HUNG Q

ART UNIT	PAPER NUMBER
----------	--------------

2168

DATE MAILED: 10/31/2005

Please find below and/or attached an Office communication concerning this application or proceeding.



UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450
www.uspto.gov

MAILED

OCT 31 2005

Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 09/422,998
Filing Date: October 21, 1999
Appellant(s): HEPNER ET AL.

Jody C. Bishop Reg. No. 44,034
HEWLETT PACKARD COMPANY
INTELLECTUAL PROPERTY ADMINISTRATION
3404 E HARMONY ROAD
P.O. BOX 272400
FORT COLLINS, CO 80528-9599
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 04/25/2005.

(1) *Real Party in Interest*

A statement identifying the real party in interest is contained in the brief.

(2) *Related Appeals and Interferences*

A statement identifying the related appeals and interferences which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief.

(3) *Status of Claims*

The statement of the status of the claims contained in the brief is correct.

(4) *Status of Amendments After Final*

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) *Summary of Invention*

The summary of invention contained in the brief is correct.

(6) *Issues*

The appellant's statement of the issues in the brief is correct.

(7) Grouping of Claims

The rejection of claims 1-5, 7-14, 16-18 and 20-22 stand or fall together because appellant's brief does not include a statement that this grouping of claims does not stand or fall together and reasons in support thereof. See 37 CFR 1.192(c)(7).

(8) Claims Appealed

The claims on appeal are claims 1-5, 7-14, 16-18 and 20-22. The Appendix only has claims 1-5 and 7-14. A copy of the claims on appeal is attached to the Examiner Answer.

(9) Prior Art of Record

6,182,249	Wookey et al.	01-2001
-----------	---------------	---------

Sybase, "Transact-SQL User's Guide", Chapters 1, 2, 7, 8 and 14.

(10) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the

Art Unit: 2162

invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1-5, 8-13, 16-18 and 20-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wookey et al. [USP 6,182,249 B1].

Regarding claim 1, Wookey teaches a method of monitoring the state and generating alerts indicating predefined conditions exist in a computer system (Abstract and Col. 1, Lines 20-23). The method comprising:

- *receiving by a reporting application, which includes computer-executable software code stored to a computer-readable medium, a request from a client to notify said client of a condition of an attribute of a system, wherein said request comprises information specifying a query for system attribute* (As illustrated at FIG. 1, the computer system 102 is monitored by computer system 100. As taught by Wookey, the diagnostic tests as shown in Table 1 or 2 are run periodically on the computer system 102 under

the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). Output data can be used to determine an existence of condition of system attribute, e.g., the amount of disk free over time can be monitored by looking at output of the *dt* test over a period of time (Col. 11, Lines 28-30), and can be analyzed to present alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 56-59). The existence of the alerts is communicated to, e.g., a support engineer (Col. 16, Lines 41-42). As further disclosed by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20)).

As defined in the Microsoft Press Computer Dictionary 3rd edition,

Query

1. The process of extracting data from a database and presenting it for use.
2. A specific set of instructions for extracting particular data repetitively.

As seen, the monitor control software as *reporting application* uses a periodically running test as *query* to extract system attribute for monitoring system attribute's condition. In order to receive an alert of a particular system attribute, e.g., alert indicates the amount of disk free is below a threshold, a support engineer at the monitoring computer system 100 has to make a request of a specific test to be enabled, e.g., *dt* test, by selecting. The enable *request from* the support engineer as *client comprises* test name as *information specifying a query*,

obviously, is *received by* the monitor control software as *reporting application*, which enables the test or *query as specified by the enable request* by the test name to extract system attributes as *queries the system for* monitoring system attribute's condition as *existence of said condition of said attribute*);

- *receiving by said reporting application raw data from said system* (Diagnostic tests are run under control of monitor control software, and the raw data resulted from those test are provided to monitoring computer system 100 (Col. 4, Lines 4-18). As seen, in order to provide raw data to monitoring computer system 100, the raw data has to be received at the monitor control software for distributing);
- *deriving said data about said system attribute to determine if said condition exists* (information from the diagnostic data are extracted (Col. 7, Lines 10-11), e.g., disk partition ID, last sector, first sector... as *system attribute* (Col. 7, Line 66-Col. 8, Line 4), for the presence of alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 57-60), e.g., an alert to determine if a particular partition has exceeded a predetermined percentage used, when the two conditions are true, e.g., partition name = /var AND percentage used $\geq 80\%$, the alert is raised (Col. 12, Lines 48-55)); and
- *upon determining that said condition exist, notifying said client of the existence of said condition* (e.g., an alert to determine if a particular partition has exceeded a predetermined percentage used, when the two conditions are true, e.g., partition name = /var AND percentage used $\geq 80\%$, the alert is raised (Col. 12, Lines 48-55), and communicated to the support engineer (Col. 16, Lines 41-42)).

Wookey does not explicitly teach the step of *using by said reporting application said query for querying said system, as specified by said request, for existence of said condition of said attribute.*

However, as taught by Wookey, the diagnostic tests as shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). Output data can be used to determine an existence of condition of system attribute, e.g., the amount of disk free over time can be monitored by looking at output of the *df* test over a period of time (Col. 11, Lines 28-30), and can be analyzed to present alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 56-59). The existence of the alerts is communicated to, e.g., a support engineer (Col. 16, Lines 41-42). As seen, the monitor control software as *reporting application* uses a periodically running test as *query* to extract system attribute for monitoring system attribute's condition. In short, the Wookey technique performs the claimed *using by said reporting application said query for querying said system as specified by said request, for existence of said condition of said attribute.*

As suggested by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). Obviously, in the knowledge of one of ordinary skill in the art, to receive an alert of a particular system attribute, e.g., alert indicates the amount of disk free is below a threshold, a support engineer has to

Art Unit: 2162

make a request of a specific test to be enabled, e.g., *dt* test, by selecting as taught by Wookey, and the test or *query is specified by enable request*.

Regarding claim 2, Wookey teaches all the claimed subject matters as discussed in claim 1, Wookey further discloses the step of *generating derived data based upon the result of said query of said system* (Col. 7, Lines 10-20 and Col. 9, Lines 42-49).

Regarding claims 3, Wookey teaches all the claimed subject matters as discussed in claims 1, Wookey further discloses *condition is a change in said attribute* (Col. 12, Lines 4-13).

Regarding claim 4, Wookey teaches a method of monitoring the state and generating alerts indicating predefined conditions exist in a computer system (Abstract and Col. 1, Lines 20-23).

- *receiving a request from a client to notify said client of a condition of an attribute of a system, wherein said request comprises information specifying a query for said system attribute* (As illustrated at FIG. 1, the computer system 102 is monitored by computer system 100. As taught by Wookey, the diagnostic tests as shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). Output data can be used to determine an existence of condition of system

attribute, e.g., the amount of disk free over time can be monitored by looking at output of the *dt* test over a period of time (Col. 11, Lines 28-30), and can be analyzed to present alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 56-59). The existence of the alerts is communicated to, e.g., a support engineer (Col. 16, Lines 41-42). As further disclosed by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20).

As defined in the Microsoft Press Computer Dictionary 3rd edition,

Query

1. *The process of extracting data from a database and presenting it for use.*
2. *A specific set of instructions for extracting particular data repetitively.*

As seen, the enable *request* of a specific test, e.g., *dt* test, *from* the support engineer as *client*, the purpose is to determine an existence of condition of system attribute for presenting an alert to the support engineer or *to notify client of a condition of an attribute of a system*. The enable *request comprises* test name as *information specifying a query, received* at the monitored system 102. The test is *a query for said system attribute*, e.g., *dt* test);

- *deriving data about said system attribute to determine if said condition exists* (information from the diagnostic data are extracted (Col. 7, Lines 10-11), e.g., disk partition ID, last sector, first sector... as *system attribute* (Col. 7, Line 66-Col. 8, Line 4), for the presence of alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 57-60), e.g., an

alert to determine if a particular partition has exceeded a predetermined percentage used, when the two conditions are true, e.g., partition name = /var AND percentage used $\geq 80\%$, the alert is raised (Col. 12, Lines 48-55)); and

- *upon determining that said condition exist, notifying said client of the existence of said condition* (e.g., an alert to determine if a particular partition has exceeded a predetermined percentage used, when the two conditions are true, e.g., partition name = /var AND percentage used $\geq 80\%$, the alert is raised (Col. 12, Lines 48-55), and communicated to the support engineer (Col. 16, Lines 41-42)).

Wookey does not explicitly teach the step of *using said query for querying said system, as specified by said request, for existence of said condition of said attribute*.

However, as taught by Wookey, the diagnostic tests as shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). Output data can be used to determine an existence of condition of system attribute, e.g., the amount of disk free over time can be monitored by looking at output of the *df* test over a period of time (Col. 11, Lines 28-30), and can be analyzed to present alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 56-59). The existence of the alerts is communicated to, e.g., a support engineer (Col. 16, Lines 41-42). As seen, the monitor control software as *reporting application* uses a periodically running test as *query* to extract system attribute for monitoring system attribute's condition. In short, the Wookey technique performs the claimed *using by said*

Art Unit: 2162

reporting application said query for querying said system as specified by said request, for existence of said condition of said attribute.

As suggested by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). Obviously, in the knowledge of one of ordinary skill in the art, to receive an alert of a particular system attribute, e.g., alert indicates the amount of disk free is below a threshold, a support engineer has to make a request of a specific test to be enabled, e.g., *dt* test, by selecting as taught by Wookey, and the test or *query is specified by enable request* by its test name.

Regarding claim 5, Wookey teaches all the claim subject matters as discussed in claim 1, Wookey further discloses *client is selected from the group consisting of a user and a client application program* (FIG. 2).

Regarding claim 8, Wookey teaches all the claimed subject matters as discussed in claim 1, Wookey further discloses *information specifying a query for said system attribute comprises multiple transactions bracketed together* (As disclosed by Wookey, the tests can be selectively enabled or disable according to the monitor system (Col. 16, Lines 19-20). Wookey further discloses the tests are run at a particular time period (Col. 5, Lines 45-46). As seen, in order to specify a test or query in an enable request, a support engineer have to select a test for enabling, also specify a run time for the test. Selecting a test for enabling, and specifying a run time are *multiple transactions bracketed together*).

Regarding claim 9, Wookey teaches all the claimed subject matters as discussed in claim 1, Wookey further discloses *multiple transactions bracketed together, wherein upon determining that such bracketed condition exist, notifying said client of the existence of such bracketed conditions* (As disclosed by Wookey, a spot alert indicates that a particular value of a system component has exceeded a threshold value, e.g., the root partition of a disk exceeds 99% (Col. 12, Lines 2-7). Two conditions must be true to raise the alert, if partition name = /var AND percentage used $\geq 99\%$ (Col. 12, Lines 48-55). As seen, two conditions are bracketed by AND operator, and if the two condition are true, the alert will be raised).

Regarding claim 10, Wookey teach all the claimed subject matters as discussed in claim 9, Wookey further discloses *the multiple changes are bracketed together, wherein upon determining that such bracketed changes exist, notifying said client of the existence of such bracketed changes* (As disclosed by Wookey, a predictive alert analyzes historical and current data to identify trends (Col. 12, Lines 14-15), e.g., detecting increasing disk usage and predict the problem before the threshold of 99 % is reached (Col. 12, Lines 27-43). As seen, a current change in disk usage, and a historical change are bracketed for comparing to raise an alert before the threshold of 99 % is reached).

Regarding claim 11, Wookey teaches all the claim subject matters as discussed in claim 1, Wookey further discloses *client is a graphical user interface (GUI) that displays information to a human user* (Col. 16, Lines 41-58).

Regarding claim 12, Wookey teaches all the claim subject matters as discussed in claim 11, Wookey further discloses the step of *deriving data to determine if a condition of said one or more attributes exists such that the GUI should redraw the graphics displaying said information about said one or more attributes* (Col. 16, Line 41-Col. 17, Line 18).

Regarding claim 13, Wookey teaches a computer program for monitoring the state and generating alerts indicating predefined conditions exist in a computer system (Abstract, Col. 1, Lines 20-23 and Col. 40, Lines 57-60). The computer program comprises the steps:

- *receiving from a client a request to notify said client of a condition of an attribute of a system, wherein said request comprises information specifying a query for said system attribute* (As illustrated at FIG. 1, the computer system 102 is monitored by computer system 100. As taught by Wookey, the diagnostic tests as shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). Output data can be used to determine an existence of condition of system attribute, e.g., the amount of disk free over time can be monitored by looking at output of the *df* test over a period of time (Col. 11, Lines 28-30), and can be analyzed to present alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 56-59). The existence of the alerts is communicated to, e.g., a support engineer (Col. 16,

Lines 41-42). As further disclosed by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20).

As defined in the Microsoft Press Computer Dictionary 3rd edition,

Query

1. *The process of extracting data from a database and presenting it for use.*
2. *A specific set of instructions for extracting particular data repetitively.*

As seen, the enable *request* of a specific test, e.g., *dt* test, *from* the support engineer as *client*, the purpose is to determine an existence of condition of system attribute for presenting an alert to the support engineer or *to notify client of a condition of an attribute of a system*. The enable *request comprises* test name as *information specifying a query, received* at the monitored system 102. The test is a *query for said system attribute, e.g., dt test*);

- *deriving data about said system attribute* (information from the diagnostic data are extracted (Col. 7, Lines 10-11), e.g., disk partition ID, last sector, first sector... as *system attribute* (Col. 7, Line 66-Col. 8, Line 4));
- *determining from said derived data if said condition exist* (alerts are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 57-60), e.g., an alert to determine if a particular partition has exceeded a predetermined percentage used, when the two conditions are true, e.g., partition name = /var AND percentage used $\geq 80\%$, the alert is raised (Col. 12, Lines 48-55));

- *upon determining that said condition exist, notifies said client of the existence of said condition* (e.g., an alert to determine if a particular partition has exceeded a predetermined percentage used, when the two conditions are true, e.g., partition name = /var AND percentage used $\geq 80\%$, the alert is raised (Col. 12, Lines 48-55), and communicated to the support engineer (Col. 16, Lines 41-42)).

Wookey does not explicitly teach the step of *querying said system as specified by said request*.

However, as suggested by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). As discussed above with respect to the step of receiving from a client a request, the monitor control software uses a periodically running test as *query* to extract system attribute for monitoring system attribute's condition, e.g., the amount of disk free over time can be monitored by looking at output of the *df* test over a period of time (Col. 11, Lines 28-30). Thus, before the test is run to extract system attribute as *querying said system*, obviously, the test is *specified by the request*, e.g., test name.

Regarding claim 16, Wookey teaches all the claimed subject matters as discussed in claim 13, Wookey further discloses *condition is a change in said attribute* (Col. 12, Lines 4-13).

Regarding claim 17, Wookey teaches all the claimed subject matters as discussed in claim 13, Wookey further discloses *multiple conditions bracketed together*,

Art Unit: 2162

wherein upon determining that such bracketed conditions exist, notifying said client of the existence of such bracketed conditions (As disclosed by Wookey, a predictive alert analyzes historical and current data to identify trends (Col. 12, Lines 14-15), e.g., detecting increasing disk usage and predict the problem before the threshold of 99 % is reached (Col. 12, Lines 27-43). As seen, a current change in disk usage, and a historical change are bracketed for comparing to raise an alert before the threshold of 99 % is reached).

Regarding claim 18, Wookey teaches a system for monitoring the state and generating alerts indicating predefined conditions exist in a computer system (Abstract, Col. 1, Lines 20-23 and Col. 40, Lines 57-60). The system comprises:

- *means for storing a reporting application* (monitoring control software as *reporting application* is stored in a computer (Col. 4, Lines 1-6));
- *a means for executing said reporting application* (monitoring control software as *reporting application* is executed by the computer (Col. 4, Lines 1-6));
- *reporting application includes computer executable software code for receiving from a client application program a request to notify said client application program of a condition of an attribute of a system, said request comprises information specifying a query for said system attribute* (As illustrated at FIG. 1, the computer system 102 is monitored by computer system 100. As taught by Wookey, the diagnostic tests as shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). Output data can be used to determine an existence of condition of

system attribute, e.g., the amount of disk free over time can be monitored by looking at output of the *dt* test over a period of time (Col. 11, Lines 28-30), and can be analyzed to present alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 56-59). The existence of the alerts is communicated to, e.g., a support engineer (Col. 16, Lines 41-42). As further disclosed by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20).

As defined in the Microsoft Press Computer Dictionary 3rd edition,

Query

1. *The process of extracting data from a database and presenting it for use.*
2. *A specific set of instructions for extracting particular data repetitively.*

As seen, monitor control software as *reporting application* implies *computer executable software code*. The monitor control software *receives* an enabled *request* for a specific test for running on the computer system 102. The enable *request* of a specific test, e.g., *dt* test, *from* the technique of user selecting as taught by Wookey implies a Graphical User Interface to facilitate selection as *client application program*. The purpose is to determine an existence of condition of system attribute for presenting an alert to the support engineer or *to notify client application program of a condition of an attribute of a system*. The enable *request* comprises test name as *information specifying a query*. The test is *a query for said system attribute*, e.g., *dt* test);

- *determining if said condition exist* (alerts are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 57-60), e.g., an alert to determine if a particular partition has exceeded a predetermined percentage used, when the two conditions are true, e.g., partition name = /var AND percentage used $\geq 80\%$, the alert is raised (Col. 12, Lines 48-55));
- *upon determining that said condition exist, notifies said client application program of the existence of said condition* (e.g., an alert to determine if a particular partition has exceeded a predetermined percentage used, when the two conditions are true, e.g., partition name = /var AND percentage used $\geq 80\%$, the alert is raised (Col. 12, Lines 48-55), and communicated to the support engineer (Col. 16, Lines 41-42)).

Wookey does not explicitly teach the step of *querying said system as specified by said request*.

As suggested by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). As discussed above with respect to the step of receiving from a client a request, the monitor control software uses a periodically running test as *query* to extract system attribute for monitoring system attribute's condition, e.g., the amount of disk free over time can be monitored by looking at output of the *df* test over a period of time (Col. 11, Lines 28-30). Thus, before the test is run to extract system attribute as *querying said system*, obviously, the test is *specified by the request*, e.g., test name.

Regarding claim 20, Wookey teaches all the claim subject matters as discussed in claim 18, Wookey further discloses *multiple nodes, wherein at least one of said nodes is executing said reporting application* (FIG. 1).

Regarding claim 21, Wookey teaches all the claim subject matters as discussed in claim 13, Wookey further discloses the step of *periodically querying the system* (Col. 3, Line 63-Col. 4, Line 26).

Regarding claim 22, Wookey teaches all the claimed subject matters as discussed in claim 18, Wookey further discloses the step of *monitoring system to determine if said condition exist* (Col. 12, Lines 44-55).

Claims 7 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wookey et al. [USP 6,182,249] in view of Sybase [Transact-SQL User's Guide, Copyright 1996].

Regarding claim 7, Wookey teaches a method of monitoring the state and generating alerts indicating predefined conditions exist in a computer system (Abstract and Col. 1, Lines 20-23).

- receiving a request from a client to notify said client of a condition of an attribute of a system, wherein said request comprises information specifying a query for said system attribute (As illustrated at FIG. 1, the computer system 102 is monitored by computer system 100. As taught by Wookey, the diagnostic tests

as shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). Output data can be used to determine an existence of condition of system attribute, e.g., the amount of disk free over time can be monitored by looking at output of the **dt** test over a period of time (Col. 11, Lines 28-30), and can be analyzed to present alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 56-59). The existence of the alerts is communicated to, e.g., a support engineer (Col. 16, Lines 41-42). As further disclosed by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20).

As defined in the Microsoft Press Computer Dictionary 3rd edition,

Query

1. *The process of extracting data from a database and presenting it for use.*
2. *A specific set of instructions for extracting particular data repetitively.*

As seen, the enable *request* of a specific test, e.g., **dt** test, *from* the support engineer as *client*, the purpose is to determine an existence of condition of system attribute for presenting an alert to the support engineer or *to notify client of a condition of an attribute of a system*. The enable *request comprises* test name as *information specifying a query, received* at the monitored system 102. The test is *a query for said system attribute, e.g., dt test*);

- *deriving data about said system attribute* (information from the diagnostic data are extracted (Col. 7, Lines 10-11), e.g., disk partition ID, last sector, first sector... as *system attribute* (Col. 7, Line 66-Col. 8, Line 4))

to determine if said condition exists (alerts are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 57-60), e.g., an alert to determine if a particular partition has exceeded a predetermined percentage used, when the two conditions are true, e.g., partition name = /var AND percentage used $\geq 80\%$, the alert is raised (Col. 12, Lines 48-55));

- *upon determining that said condition exists, notifying said client of the existence of said condition* (e.g., an alert to determine if a particular partition has exceeded a predetermined percentage used, when the two conditions are true, e.g., partition name = /var AND percentage used $\geq 80\%$, the alert is raised (Col. 12, Lines 48-55), and communicated to the support engineer (Col. 16, Lines 41-42)).

Wookey does not explicitly teach the claimed *querying said system as specified by said request*, and fails to disclose *an SQL query comprises an SQL view* is used to specify a query.

However, as suggested by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). As discussed above with respect to the step of receiving from a client a request, the monitor control software uses a periodically running test as *query* to extract system attribute for monitoring system attribute's condition, e.g., the amount of disk free over time can be monitored by looking at output of the *df* test over a period of time (Col. 11, Lines 28-30). Thus, before

Art Unit: 2162

the test is run to extract system attribute as *querying said system*, obviously, the test is *specified by the request*, e.g., test name.

Sybase teaches SQL as a high-level language includes commands for retrieving data from a database, creating database object and other functions (Sybase, Chapter 1: Introduction, Overview). As shown in Chapter 1 is the method of creating SQL statements by using select command. As shown in Chapter 14 is the method of creating trigger conditions by using SQL statements. Sybase further discloses *SQL query comprises an SQL view* (Sybase, Chapter 8, Views: Limiting access to Data, Creating Views).

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to use SQL as taught by Sybase to implement the test in order to issue an alert indicating predefined condition exist in a computer system.

Regarding claim 14, Wookey teaches all the claimed subject matters as discussed in claim 13, but fails to teach *information specifying a query for said system attribute is an SQL query*. Sybase teaches SQL as a high level language for relational database system and using query as a request for retrieval of data by using the select command (Sybase, Chapter 1: Introduction, Overview and Queries, Data Modification). Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to use SQL for implementing the test in order to issue an alert indicating predefined condition exist in a computer system.

(11) Response to Argument

A. RESPONSE TO ARGUMENTS WITH RESPECT TO THE FIRST GROUND OF REJECTION

1. CLAIMS 1-3, 5, 11 AND 12

(a) As argued by appellants regarding to a lack of motivation to modify at page 6, lines 10-26 of the Brief:

Wookey does not teach "using by said reporting application said query for querying said system as specified by said request, for existence of said condition of said attribute."... The proposed modification of Wookey is improper because the motivation provided is incorrect.

Examiner respectfully disagrees. A prima facie case of obviousness under 35 U.S.C. § 103 (a) was established for the claimed limitation as set forth in the Final Action 01/11/05.

As defined in the Microsoft Press Computer Dictionary 3rd edition, a "query" is:

- Query*
- 1. The process of extracting data from a database and presenting it for use.*
 - 2. A specific set of instructions for extracting particular data repetitively.*

As taught by Wookey, the diagnostic tests shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-

38). Output data can be used to monitor system attributes, e.g., monitoring the amount of free disk space by looking at the output of the *dt* test over a period of time (Col. 11, Lines 28-30), and can be analyzed to present alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 56-59), e.g., an alert to indicate a particular partition has exceeded a predetermined percentage used. If the conditions are true, the alert is raised (Col. 12, Lines 48-55). The existence of the alerts is communicated to, e.g., a support engineer (Col. 16, Lines 41-42). As seen, the monitor control software operates as a *reporting application* that uses a periodically run test as a *query* to extract system attributes in order to determine if a system attribute's condition exists. In short, the Wookey technique, as discussed above performs the claimed *using by said reporting application said query for querying said system for existence of said condition of said attribute*.

As suggested by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). Because, as it was well known to one of ordinary skill in the art, in order to enable a specific test, a test name has to be specified, e.g., *dt* test, by selecting as taught by Wookey, and the test or *query is specified by the enable request* via test name.

As supportive evidence for examiner's conclusion, the examiner respectfully refers appellants to USP 6,023,507, which is incorporated by reference by the Wookey patent (Wookey, Col. 1, Lines 5-17). As illustrated in USP 6,023,507, the monitoring software on the monitored system includes an

administrator tool (Col. 5, Lines 56-59). Administrator tool is a Graphical User Interface (Col. 8, Lines 31-37) that has an edit mode (Col. 13, Lines 12-13) presenting an administrator with a list of tests and allowing creation of test schema by selectively enabling tests and scheduling those selectively enabled test (FIG. 9, Col. 50, Lines 50-54).

(b) As argued by appellants regarding to Failure to Teach or Suggest Every Limitation from page 6, line 27 to page 8, line 16:

... Wookey does not teach or suggest every element of claim 1. As discussed further below, Wookey does not teach or suggest a reporting application that receives a request from a client and queries a system as specified by that request... There is no indication in the teaching of Wookey that the monitoring system has any control over the tests that are run on the monitored system or that it requests that any tests be run.

... it can be seen that Wookey does not teach or suggest a reporting application that receives a request from a client and queries a system as specified by that request because the monitoring system sends no request to the monitored system and has no control over the tests that are run.. Specifically, claim 1 recites, in part, "using by said reporting application said query for querying said system, as specified by said request, for existence of said condition of said attribute." Wookey, as modified, does not teach or suggest at least the above-recited feature of claim 1.

...

... claim 1 requires that the request be "from a client." The Final Action's rejection of claim 1 fails to show "querying said system, as specified by said request," where the request is from a client because Wookey's computer system 100 (the monitoring system) is passive with regard to the diagnostic tests that are run on the monitored system.

...

... there is no indication in Wookey that computer system 100 has any control or influence over the operation of the monitor control software or the specific tests that are performed. This is illustrated by the passage at column 16, lines 16-34, which reads, in

part, "However, the tests can be selectively enabled (or disabled) according to the monitored system" (emphasis added). Thus, the rejection must fail because it does not show that Wookey teaches or suggests, "querying said system, as specified by said request." Accordingly, Wookey, as modified, does not teach or suggest the above-recited feature of claim 1.

Examiner respectfully disagrees because *the passage at column 16, lines 16-34, which reads, in part, "However, the tests can be selectively enabled (or disabled) according to the monitored system implies computer system 100 has control over the operation of the specific tests that are performed, and Wookey discloses a reporting application that receives a request from a client and queries a system as specified by that request, for existence of said condition of said attribute.*

As illustrated at FIG. 1, the computer system 102 is monitored by computer system 100. As taught by Wookey, the diagnostic tests shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in the monitored system 102 (Col. 5, Lines 36-38). Output data can be used to monitor system attributes, e.g., monitoring the amount of free disk space by looking at the output of the *df* test over a period of time (Col. 11, Lines 28-30), and can be analyzed to present alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 56-59), e.g., an alert to determine a particular partition has exceeded a predetermined percentage used. If the conditions are true, the alert is raised (Col. 12, Lines 48-55). The existence of the

alerts is communicated to, e.g., a support engineer (Col. 16, Lines 41-42). As further disclosed by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). As seen, the monitor control software functions as *reporting application* that uses a periodically run test as a *query* to extract system attributes. In order to receive an alert of a particular system attribute, e.g., an alert to determine a particular partition has exceeded a predetermined percentage used, a support engineer at the monitoring computer system 100 has to make a request of a specific test to be enabled, e.g., **dt** test, by selecting. The enable *request from* the support engineer as *client*, is *received by* the monitor control software as *reporting application*. In order to enable a test to get the test data, e.g., the amount of free disk space over time, or *queries the system*, test name has to be specified in the enable request, e.g., **dt** test, and the test or *query is specified by the enable request*. The test or *query* is used for determining system attribute's condition in the form of alert, e.g., alert to determine a particular partition has exceeded a predetermined percentage used, as *existence of said condition of said attribute*.

As supportive evidence for examiner's conclusions, the examiner respectfully refers appellants to USP 6,023,507, which is incorporated by reference by the Wookey patent (Wookey, Col. 1, Lines 5-17). As illustrated in USP 6,023,507, the monitoring software on the monitored system includes an administrator tool, and communications software (Col. 5, Lines 56-59) for communicating with monitoring system (Col. 5, Line 66-Col. 6, Line 2).

Administrator tool is a Graphical User Interface (Col. 8, Lines 31-37) that has an edit mode (Col. 13, Lines 12-13) presenting an administrator with a list of tests and allowing creation of test schema by selectively enabling tests and scheduling those selectively enabled test (FIG. 9, Col. 50, Lines 50-54).

(c) As argued by appellants at page 8, lines 17-22:

Dependent claims 2, 3, 5, and 8-12 each depends from independent claim 1 and, thus, inherit all of the limitations of independent claim 1... It is respectfully submitted that dependent claims 2, 3, 5, and 8-12 are allowable at least because of their dependence from claim 1 for the reasons discussed above.

Examiner respectfully disagrees. Dependent claims 2, 3, 5, and 8-12 are unpatentable over Wookey for the reasons discussed above.

2. CLAIM 4

(a) As argued by appellants regarding to a lack of motivation to modify from page 8, line 26 to page 9, line 12 of the Brief:

The Final Action admits that Wookey does not teach "using said query for querying said systems as specified by said request, for existence of said condition of said attribute." Final Action at 17. The Final Action then asserts that it would be obvious to modify Wookey to include this feature. Id. at 18. The proposed modification of Wookey is improper because the motivation provided is incorrect.

Examiner respectfully disagrees. A prima facie case of obviousness under 35 U.S.C. § 103 (a) was established for the claimed limitation as set forth in the Final Action 01/11/05.

As defined in the Microsoft Press Computer Dictionary 3rd edition, a "query" is:

- Query*
- 1. The process of extracting data from a database and presenting it for use.*
 - 2. A specific set of instructions for extracting particular data repetitively.*

As taught by Wookey, the diagnostic tests shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). Output data can be used to monitor system attributes, e.g., monitoring the amount of free disk space by looking at the output of the *df* test over a period of time (Col. 11, Lines 28-30), and can be analyzed to present alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 56-59), e.g., an alert to indicate a particular partition has exceeded a predetermined percentage used. If the conditions are true, the alert is raised (Col. 12, Lines 48-55). The existence of the alerts is communicated to, e.g., a support engineer (Col. 16, Lines 41-42). As seen, the monitor control software operates as a *reporting application* that uses a periodically run test as a *query* to extract system attributes in order to determine if a system attribute's

condition exists. In short, the Wookey technique as discussed performs the claimed *using said query for querying said system for existence of said condition of said attribute.*

As suggested by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). Because, as it was well known to one of ordinary skill in the art, in order to enable a specific test, a test name has to be specified, e.g., *dt* test, by selecting as taught by Wookey, and the test or *query is specified by the enable request* via test name.

As supportive evidence for examiner's conclusion, the examiner respectfully refers appellants to USP 6,023,507, which is incorporated by reference by the Wookey patent (Wookey, Col. 1, Lines 5-17). As illustrated in USP 6,023,507, the monitoring software on the monitored system includes an administrator tool (Col. 5, Lines 56-59). Administrator tool is a Graphical User Interface (Col. 8, Lines 31-37) that has an edit mode (Col. 13, Lines 12-13) presenting an administrator with a list of tests and allowing creation of test schema by selectively enabling tests and scheduling those selectively enabled test (FIG. 9, Col. 50, Lines 50-54).

(b) As argued by appellants regarding to Failure to Teach or Suggest Every Limitation from page 9, line 15 to page 11, line 2:

Wookey does not teach or suggest every element of claim 4. As discussed further below, Wookey does not teach or suggest receiving a request from a client and querying a system as specified by that request...There is no indication in the teaching

of Wookey that the monitoring system has any control over the tests that are run on the monitored system or that it requests that any tests be run.

...

it can be seen that Wookey does not teach or suggest receiving a request from a client and querying a system as specified by that request because the monitoring system sends no request to the monitored system and has no control over the tests that are run. Specifically, claim 4 recites, in part, "using said query for querying said systems as specified by said request, for existence of said condition of said attribute." Wookey, as modified, does not teach or suggest at least the above-recited feature of claim 4.

...

... claim 4 requires that the request be "from a client." The rejection of claim 4 fails to show "querying said systems as specified by said request," where the request is from a client because Wookey's computer system 100 (the monitoring system) is passive with regard to the diagnostic tests that are run on the monitored system.

...

... there is no indication in Wookey that computer system 100 has any control or influence over the operation of the monitor control software or the specific tests that are performed. This is illustrated by the passage at column 16, lines 16-34, which reads, in part, "However, the tests can be selectively enabled (or disabled) according to the monitored system" (emphasis added). Thus, the rejection must fail because it does not show that Wookey teaches or suggests, "querying said systems as specified by said request. Accordingly, Wookey, as modified, does not teach or suggest the above-recited feature of claim 4.

Examiner respectfully disagrees because *the passage at column 16, lines 16-34, which reads, in part, "However, the tests can be selectively enabled (or disabled) according to the monitored system implies computer system 100 has control over the operation of the specific tests that are performed, and Wookey discloses the claimed using said query for*

querying said systems as specified by said request, for existence of said condition of said attribute, where the request is from a client.

As illustrated at FIG. 1, the computer system 102 is monitored by computer system 100. As taught by Wookey, the diagnostic tests shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in the monitored system 102 (Col. 5, Lines 36-38). Output data can be used to monitor system attributes, e.g., monitoring the amount of free disk space by looking at the output of the *df* test over a period of time (Col. 11, Lines 28-30), and can be analyzed to present alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 56-59), e.g., an alert to indicate a particular partition has exceeded a predetermined percentage used. If the conditions are true, the alert is raised (Col. 12, Lines 48-55). The existence of the alerts is communicated to, e.g., a support engineer (Col. 16, Lines 41-42). As further disclosed by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). As seen, in order to receive an alert of a particular system attribute, e.g., alert indicates the amount of free disk space is below a threshold, a support engineer at the monitoring computer system 100 has to make a request of a specific test to be enabled, e.g., *df* test, by selecting, and the enable *request* is *from* the support engineer as *client*. In order to enable a test to get the test data, e.g., the amount of free disk

space over time, or *queries the system*, the test name has to be specified in the enable request, e.g., **dt** test, and the test or *query* is *specified by the enable request* via test name. The test or *query* is used *for determining system attribute's* condition in the form of alert, e.g., alert to determine a particular partition has exceeded a predetermined percentage used, as *existence of said condition of said attribute*.

As supportive evidence for examiner's conclusion, the examiner respectfully refers appellants to USP 6,023,507, which is incorporated by reference by the Wookey patent (Wookey, Col. 1, Lines 5-17). As illustrated in USP 6,023,507, the monitoring software on the monitored system includes an administrator tool, and communications software (Col. 5, Lines 56-59) for communicating with monitoring system (Col. 5, Line 66-Col. 6, Line 2). Administrator tool is a Graphical User Interface (Col. 8, Lines 31-37) that has an edit mode (Col. 13, Lines 12-13) presenting an administrator with a list of tests and allowing creation of test schema by selectively enabling tests and scheduling those selectively enabled test (FIG. 9, Col. 50, Lines 50-54).

3. CLAIM 8

(a) As argued by appellants regarding to a lack of motivation to modify at page 11, lines 7-9 of the Brief:

As explained above, the motivation provided for modifying Wookey, as proposed is improper. The rejection of claim 8 adds nothing to correct this deficiency. Accordingly, the rejection of claim 8 must fail for lack of motivation to modify Wookey.

Examiner respectfully disagrees. Dependent claim 8 is unpatentable over Wookey for the reasons discussed above with respect to claim 1.

(b) As argued by appellants regarding to Failure to Teach or Suggest Every Limitation at page 11, lines 11-20:

Claim 8 recites, in part, "wherein said information specifying a query for said system attribute comprises multiple transactions bracketed together." Wookey does not teach or suggest at least this feature... Claim 8 is also allowable because of its dependence from claim 1, which, as explained above, is allowable.

Examiner respectfully disagrees. The Wookey reference, taken as a whole, discloses the features of claim 8. As disclosed by Wookey, the tests can be selectively enabled or disabled according to the monitor system (Col. 16, Lines 19-20). Wookey further discloses the tests are run at a particular time period (Col. 5, Lines 45-46). As seen, in order to specify a test or query in an enable request, a support engineer has to select a test for enabling, also specify a run time for the test. Selecting a test for enabling, and specifying a run time are *multiple transactions bracketed together*.

As supportive evidence for examiner's conclusion, the examiner respectfully refers appellants to USP 6,023,507, which is incorporated by reference by the Wookey patent (Wookey, Col. 1, Lines 5-17). As illustrated at USP 6,023,507 FIGS. 9 and 11, a GUI is used by a system administrator to make an enable request and specify a run time for a test.

4. CLAIM 9

(a) As argued by appellants regarding to a lack of motivation to modify at page 11, lines 25-27 of the Brief:

As explained above, the motivation provided for modifying Wookey, as proposed is improper. The rejection of claim 9 adds nothing to correct this deficiency. Accordingly, the rejection of claim 9 must fail for lack of motivation to modify Wookey.

Examiner respectfully disagrees. Dependent claim 9 is unpatentable over Wookey for the reasons discussed above with respect to claim 1.

(b) As argued by appellants regarding to Failure to Teach or Suggest Every Limitation at page 12, lines 2-11:

Claim 9 recites, in part, "multiple conditions bracketed together, wherein upon determining that such bracketed conditions exist, notifying said client of the existence of such bracketed conditions." Wookey does not teach or suggest at least this feature. The Final Action cites the passage in Wookey at column 15, lines 16-54 as teaching the feature (see Final Action at 19); however, such assertion is incorrect... Accordingly, the above-recited feature of claim 9 is not taught or suggested by Wookey. Claim 9 is also allowable because of its dependence from claim 1, which, as explained above, is allowable.

Examiner respectfully disagrees. The Wookey reference, taken as a whole, discloses the features of claim 9. As disclosed by Wookey, a spot alert indicates that a particular value of a system component has exceeded a threshold value, e.g., the root partition of a disk exceeds 99% (Col. 12, Lines 2-7). Two conditions must be true to raise the alert, if partition name = /var AND

percentage used $\geq 99\%$ (Col. 12, Lines 48-55). As seen, two conditions are bracketed by an AND operator, and if the two conditions are true, the alert will be raised.

5. CLAIM 10

(a) As argued by appellants regarding to a lack of motivation to modify at page 12, lines 16-18 of the Brief:

As explained above, the motivation provided for modifying Wookey, as proposed is improper. The rejection of claim 10 adds nothing to correct this deficiency. Accordingly, the rejection of claim 10 must fail for lack of motivation to modify Wookey.

Examiner respectfully disagrees. Dependent claim 10 is unpatentable over Wookey for the reasons discussed above with respect to claim 1.

(b) As argued by appellants regarding to Failure to Teach or Suggest Every Limitation at page 12, lines 20-29:

Claim 10 recites, in part, "multiple changes bracketed together, wherein upon determining that such bracketed changes exist, notifying said client of the existence of such bracketed changes." Wookey does not teach or suggest at least this feature. The Final Action cites the passage in Wookey at column 15, lines 16-54 as teaching the feature (see Final Action at 18); however, such assertion is incorrect... Claim 10 is also allowable because of its dependence from claim 1, which, as explained above, is allowable.

Examiner respectfully disagrees. The Wookey reference, taken as a whole, discloses the features of claim 9. As disclosed by Wookey, a predictive

alert analyzes historical and current data to identify trends (Col. 12, Lines 14-15), e.g., detecting increasing disk usage and predict the problem before the threshold of 99 % is reached (Col. 12, Lines 27-43). As seen, a current change in disk usage, and a historical change are bracketed for comparing to raise an alert before the threshold of 99 % is reached.

6. CLAIMS 13, 16 AND 21

(a) As argued by appellants regarding to a lack of motivation to modify at page 13, lines 5-9 of the Brief:

The Final Action admits that Wookey does not teach "computer executable software code for querying said system as specified by said request." Final Action at 21. The Final Action then asserts that it would be obvious to modify Wookey to include this feature. Id. at 22. The proposed modification of Wookey is improper because the motivation provided is incorrect.

Examiner respectfully disagrees. A prima facie case of obviousness under 35 U.S.C. § 103 (a) was established for the claimed limitation as set forth in the Final Action 01/11/05.

As defined in the Microsoft Press Computer Dictionary 3rd edition, a "query" is:

- Query*
- 1. The process of extracting data from a database and presenting it for use.*
 - 2. A specific set of instructions for extracting particular data repetitively.*

As taught by Wookey, JAVA is used for the implementation (Col. 8, Lines 8-10). The diagnostic tests shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). As seen, the JAVA implemented test for extracting software and hardware components in monitored system 102 indicates *computer executable software code for querying said system*.

As suggested by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). Because, as it was well known to one of ordinary skill in the art, in order to enable a specific test, a test name has to be specified by selecting as taught by Wookey, and the test or *query is specified by the enable request via test name*.

As supportive evidence for examiner's conclusion, the examiner respectfully refers appellants to USP 6,023,507, which is incorporated by reference by the Wookey patent (Wookey, Col. 1, Lines 5-17). As illustrated in USP 6,023,507, the monitoring software on the monitored system includes an administrator tool (Col. 5, Lines 56-59). Administrator tool is a Graphical User Interface (Col. 8, Lines 31-37) that has an edit mode (Col. 13, Lines 12-13) presenting an administrator with a list of tests and allowing creation of test schema by selectively enabling tests and scheduling those selectively enabled test (FIG. 9, Col. 50, Lines 50-54).

(b) As argued by appellants regarding to Failure to Teach or Suggest

Every Limitation from page 13, line 21 to page 15, line 9:

Wookey does not teach or suggest every element of claim 13. As discussed further below, Wookey does not teach or suggest code in a reporting application that receives a request from a client and queries a system as specified by that request... There is no indication in the teaching of Wookey that the monitoring system has any control over the tests that are run on the monitored system or that it requests that any tests be run...

... it can be seen that Wookey does not teach or suggest code in a reporting application that receives a request from a client and queries a system as specified by that request because the monitoring system sends no request to the monitored system and has no control over the tests that are run. Specifically, claim 13 recites, in part, "computer executable software code for querying said system as specified by said request." Wookey, as modified, does not teach or suggest at least the above-recited feature of claim 13.

...

Claim 13 requires that the request be "from a client." The rejection of claim 13 fails to show "querying said system as specified by said request," where the request is from a client because Wookey's computer system 100 (the monitoring system) is passive with regard to the diagnostic tests that are run on the monitored system.

...

... There is no indication in Wookey that computer system 100 has any control or influence over the operation of the monitor control software or the specific tests that are performed. This is illustrated by the passage at column 16, lines 16-34, which reads, in part, "However, the tests can be selectively enabled (or disabled) according to the monitored system" (emphasis added). Thus, the rejection must fail because it does not show that Wookey teaches or suggests, "querying said system as specified by said request."

Examiner respectfully disagrees because the passage at column 16, lines 16-34, which reads, in part, "However, the tests can be selectively enabled (or disabled) according to the monitored system implies computer system 100 has control over the operation of the specific

tests that are performed, and Wookey discloses code in a reporting application that receives a request from a client and queries a system as specified by that request.

As taught by Wookey, JAVA is used for the implementation (Col. 8, Lines 8-10). As illustrated at FIG. 1, the computer system 102 is monitored by computer system 100. The diagnostic tests shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). As seen, the JAVA implemented test for extracting software and hardware components in monitored system 102 performs the claimed *computer executable software code for querying said system.*

As suggested by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). Because, as it was well known to one of ordinary skill in the art, in order to enable a specific test, a test name has to be specified by user's selecting, and the test or *query is specified by the enable request* via test name from a user or *client.*

As supportive evidence for examiner's conclusion, the examiner respectfully refers appellants to USP 6,023,507, which is incorporated by reference by the Wookey patent (Wookey, Col. 1, Lines 5-17). As illustrated in USP 6,023,507, the monitoring software on the monitored system includes an administrator tool (Col. 5, Lines 56-59). Administrator tool is a Graphical User Interface (Col. 8, Lines 31-37) that has an edit mode (Col. 13, Lines 12-13)

Art Unit: 2162

presenting an administrator with a list of tests and allowing creation of test schema by selectively enabling tests and scheduling those selectively enabled test (FIG. 9, Col. 50, Lines 50-54).

(c) As argued by appellants at page 15, lines 10-15:

Dependent claims 16, 17, and 21 depend from independent claim 13 and, thus, inherit all of the limitations of independent claim 13. Thus, Wookey does not teach or suggest all claim limitations of claims 16, 17, and 21. It is respectfully submitted that dependent claims 16 and 21 are allowable at least because of their dependence from claim 13 for the reasons discussed above.

Examiner respectfully disagrees. Dependent claims 16, 17 and 21 are unpatentable over Wookey for the reasons discussed above.

7. CLAIM 17

(a) As argued by appellants regarding to a lack of motivation to modify at page 15, lines 20-22 of the Brief:

As explained above, the motivation provided for modifying Wookey, as proposed is improper. The rejection of claim 17 adds nothing to correct this deficiency. Accordingly, the rejection of claim 17 must fail for lack of motivation to modify Wookey.

Examiner respectfully disagrees. Dependent claim 17 is unpatentable over Wookey for the reasons discussed above with respect to claim 13.

(b) As argued by appellants regarding to Failure to Teach or Suggest
Every Limitation from page 15, line 24 to page 16, line 4:

Claim 17 recites, in part, "multiple changes bracketed together." Wookey does not teach or suggest at least this feature... Claim 17 is also allowable because of its dependence from claim 13, which, as explained above, is allowable.

Examiner respectfully disagrees. The Wookey reference, taken as a whole, discloses the features of claim 17. As disclosed by Wookey, a predictive alert analyzes historical and current data to identify trends (Col. 12, Lines 14-15), e.g., detecting increasing disk usage and predict the problem before the threshold of 99 % is reached (Col. 12, Lines 27-43). As seen, a current change in disk usage, and a historical change are bracketed for comparing to raise an alert before the threshold of 99 % is reached.

8. CLAIMS 18, 20 AND 22

(a) As argued by appellants regarding to a lack of motivation to modify
at page 16, lines 9-23 of the Brief:

The Final Action admits that Wookey does not teach "computer executable software code for querying said system as specified by said request." Final Action at 21 ... The proposed modification of Wookey is improper because the motivation provided is incorrect.

...

Thus, the Final Action fails to provide proper motivation for the modification, and the rejection of claims 18, 20, and 22 must fail.

Examiner respectfully disagrees. A prima facie case of obviousness under 35 U.S.C. § 103 (a) was established for the claimed limitation as set forth in the Final Action 01/11/05.

As defined in the Microsoft Press Computer Dictionary 3rd edition, a “query” is:

- Query*
- 1. The process of extracting data from a database and presenting it for use.*
 - 2. A specific set of instructions for extracting particular data repetitively.*

As taught by Wookey, JAVA is used for the implementation (Col. 8, Lines 8-10). The diagnostic tests shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). As seen, the JAVA implemented test for extracting software and hardware components in the monitored system 102 indicates *computer executable software code for querying said system*.

As suggested by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). Because, as it was well known to one of ordinary skill in the art, in order to enable a specific test, test name has to be specified by selecting as taught by Wookey, and the test or *query is specified by the enable request* via test name.

As supportive evidence for examiner’s conclusion, the examiner respectfully refers appellants to USP 6,023,507, which is incorporated by

reference by the Wookey patent (Wookey, Col. 1, Lines 5-17). As illustrated in USP 6,023,507, the monitoring software on the monitored system includes an administrator tool (Col. 5, Lines 56-59). Administrator tool is a Graphical User Interface (Col. 8, Lines 31-37) that has an edit mode (Col. 13, Lines 12-13) presenting an administrator with a list of tests and allowing creation of test schema by selectively enabling tests and scheduling those selectively enabled test (FIG. 9, Col. 50, Lines 50-54).

(b) As argued by appellants regarding to Failure to Teach or Suggest Every Limitation from page 16, line 25 to page 18, line 20:

Wookey does not teach or suggest code in a reporting application that receives a request from a client and queries a system as specified by that request... There is no indication in the teaching of Wookey that the monitoring system has any control over the tests that are run on the monitored system or that it requests that any tests be run.

... it can be seen that Wookey does not teach or suggest code in a reporting application that receives a request from a client and queries a system as specified by that request because the monitoring system sends no request to the monitored system and has no control over the tests that are run. Specifically, claim 18 recites, in part, "computer executable software code for querying said system as specified by said request." Wookey, as modified, does not teach or suggest at least the above-recited feature of claim 18.

...

... claim 18 requires that the request be "from a client application program." The rejection of claim 18 fails to show "querying said system as specified by said request," where the request is "from a client application program" because Wookey's computer system 100 (the monitoring system) is passive with regard to the diagnostic tests that are run on the monitored system.

...

... there is no indication in Wookey that computer system 100 has any control or influence over the operation of the monitor control software or the specific tests that are performed. This

is illustrated by the passage at column 16, lines 16-34, which reads, in part, "However, the tests can be selectively enabled (or disabled) according to the monitored system" (emphasis added). Thus, the rejection must fail because it does not show that Wookey teaches or suggests, "querying said system as specified by said request."

Examiner respectfully disagrees because *the passage at column 16, lines 16-34, which reads, in part, "However, the tests can be selectively enabled (or disabled) according to the monitored system implies computer system 100 has control over the operation of the specific tests that are performed, and Wookey discloses code in a reporting application that receives a request from a client application program and queries a system as specified by that request.*

As taught by Wookey, JAVA is used for the implementation (Col. 8, Lines 8-10). As illustrated at FIG. 1, the computer system 102 is monitored by computer system 100. The diagnostic tests shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). As seen, the JAVA implemented test for extracting software and hardware components in monitored system 102 indicates *computer executable software code for querying said system.*

As suggested by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). Because, as it was well known to one of ordinary skill in the art, in order to enable a specific test, test name has to be specified using a Graphical User Interface to facilitate user's

selecting, and the test or *query is specified by the enable request* via the test name from a Graphical User Interface or *client application program*.

As supportive evidence for examiner's conclusion, the examiner respectfully refers appellants to USP 6,023,507, which is incorporated by reference by the Wookey patent (Wookey, Col. 1, Lines 5-17). As illustrated in USP 6,023,507, the monitoring software on the monitored system includes an administrator tool (Col. 5, Lines 56-59). Administrator tool is a Graphical User Interface (Col. 8, Lines 31-37) that has an edit mode (Col. 13, Lines 12-13) presenting an administrator with a list of tests and allowing creation of test schema by selectively enabling tests and scheduling those selectively enabled test (FIG. 9, Col. 50, Lines 50-54).

(c) As argued by appellants at page 18, lines 15-20:

Dependent claims 20 and 22 depend from independent claim 18 and, thus, inherit all of the limitations of independent claim 18. Thus, Wookey does not teach or suggest all claim limitations of claims 20 and 22.

Examiner respectfully disagrees. Dependent claims 20 and 22 are unpatentable over Wookey for the reasons discussed above.

B. RESPONSE TO ARGUMENTS WITH RESPECT TO THE SECOND
GROUND OF REJECTION

1. CLAIM 7

(a) As argued by appellants regarding to a lack of motivation to modify at page 19, lines 5-19 of the Brief:

The Final Action admits that Wookey does not teach "querying said system as specified by said request. Final Action at 25. The Final Action then asserts that it would be obvious to modify Wookey to include this feature. Id. at 25. The proposed modification of Wookey is improper because the motivation provided is incorrect.

Examiner respectfully disagrees. A prima facie case of obviousness under 35 U.S.C. § 103 (a) was established for the claimed limitation as set forth in the Final Action 01/11/05.

As defined in the Microsoft Press Computer Dictionary 3rd edition, a "query" is:

- Query
1. The process of extracting data from a database and presenting it for use.
 2. A specific set of instructions for extracting particular data repetitively.

As taught by Wookey, the diagnostic tests shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). As seen, the test for extracting software and hardware components in monitored system 102 indicates the step *querying said system*.

As suggested by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). Because, as it was

well known to one of ordinary skill in the art, in order to enable a specific test, test name has to be specified by selecting as taught by Wookey, and the test or *query* is specified by the enable request via test name.

As supportive evidence for examiner's conclusion, the examiner respectfully refers appellants to USP 6,023,507, which is incorporated by reference by the Wookey patent (Wookey, Col. 1, Lines 5-17). As illustrated in USP 6,023,507, the monitoring software on the monitored system includes an administrator tool (Col. 5, Lines 56-59). Administrator tool is a Graphical User Interface (Col. 8, Lines 31-37) that has an edit mode (Col. 13, Lines 12-13) presenting an administrator with a list of tests and allowing creation of test schema by selectively enabling tests and scheduling those selectively enabled test (FIG. 9, Col. 50, Lines 50-54).

(b) As argued by appellants regarding to Failure to Teach or Suggest Every Limitation from page 19, line 21 to page 21, line 10:

Wookey does not teach or suggest receiving a request from a client and querying a system as specified by that request... The tests that are performed are run under the control of monitor control software that is contained entirely within the monitored system. See Id. at Col. 4, lines 4-6 and figures 1 a and 1b. There is no indication in the teaching of Wookey that the monitoring system has any control over the tests that are run on the monitored system or that it requests that any tests be run...

... it can be seen that Wookey does not teach or suggest code in a reporting application that receives a request from a client and queries a system as specified by that request because the monitoring system sends no request to the monitored system and has no control over the tests that are run. Specifically, claim 7 recites, in part, "querying said system as specified by said

request." The proposed combination does not teach or suggest at least the above-recited feature of claim 7.

...

... claim 7 requires that the request be "from a client." The rejection of claim 7 fails to show "querying said system as specified by said request," where the request is from a client because Wookey's computer system 100 (the monitoring system) is passive with regard to the diagnostic tests that are run on the monitored system.

...

... there is no indication in Wookey that computer system 100 has any control or influence over the operation of the monitor control software or the specific tests that are performed. This is illustrated by the passage at column 16, lines 16-34, which reads, in part, "However, the tests can be selectively enabled (or disabled) according to the monitored system" (emphasis added). Thus, the rejection must fail because it does not show that Wookey teaches or suggests, "querying said system as specified by said request."

Examiner respectfully disagrees because *the passage at column 16, lines 16-34, which reads, in part, "However, the tests can be selectively enabled (or disabled) according to the monitored system implies computer system 100 has control over the operation of the specific tests that are performed, and Wookey discloses the step of receiving a request from a client and querying a system as specified by that request.*

As illustrated at FIG. 1, the computer system 102 is monitored by computer system 100. As taught by Wookey, the diagnostic tests shown in Table 1 or 2 are run periodically on the computer system 102 under the control of monitor control software (Col. 3, Lines 3-6). The test data contains information about the software and hardware components in monitored system 102 (Col. 5, Lines 36-38). Output data can be used to monitor system attributes, e.g., monitoring the amount of free disk space by looking at the output of the *df* test

over a period of time (Col. 11, Lines 28-30), and can be analyzed to present alerts, which are predefined conditions in the various components of the monitored computer system (Col. 11, Lines 56-59), e.g., an alert to determine a particular partition has exceeded a predetermined percentage used. If the conditions are true, the alert is raised (Col. 12, Lines 48-55). The existence of the alerts is communicated to, e.g., a support engineer (Col. 16, Lines 41-42). As further disclosed by Wookey, the test can be selectively enabled or disabled according to the monitored system (Col. 16, Lines 19-20). As seen, in order to receive an alert of a particular system attribute, e.g., alert indicates the amount of free disk space is below a threshold, a support engineer at the monitoring computer system 100 has to make a request of a specific test to be enabled, e.g., *dt* test, by selecting, and the enable *request* is *from* the support engineer as *client*. In order to enable a test to get the test data, e.g., the amount of free disk space over time, or *queries the system*, the test name has to be specified in the enable request, e.g., *dt* test, and the test or *query* is *specified by the enable request* via the test name.

As supportive evidence for examiner's conclusion, the examiner respectfully refers appellants to USP 6,023,507, which is incorporated by reference by the Wookey patent (Wookey, Col. 1, Lines 5-17). As illustrated in USP 6,023,507, the monitoring software on the monitored system includes an administrator tool, and communications software (Col. 5, Lines 56-59) for communicating with monitoring system (Col. 5, Line 66-Col. 6, Line 2).

Art Unit: 2162

Administrator tool is a Graphical User Interface (Col. 8, Lines 31-37) that has an edit mode (Col. 13, Lines 12-13) presenting an administrator with a list of tests and allowing creation of test schema by selectively enabling tests and scheduling those selectively enabled test (FIG. 9, Col. 50, Lines 50-54).

2. CLAIM 14

(a) As argued by appellants regarding to a lack of motivation to modify at page 21, lines 15-19 of the Brief:

As explained above, the motivation provided for combining Wookey and Sybase, as proposed is improper. The rejection of claim 14 adds nothing to correct this deficiency.

Examiner respectfully disagrees. Dependent claim 14 is unpatentable over Wookey for the reasons discussed above with respect to claim 13.

(b) As argued by appellants regarding to Failure to Teach or Suggest Every Limitation at page 21, lines 21-28:

Dependent claim 14 depends from independent claim 13 and, thus, inherits all of the limitations of independent claim 13. The Final Action does not rely on Sybase to teach the missing feature. Thus, the cited combination of Wookey and Sybase does not teach or suggest all claim limitations of claim 14. It is respectfully submitted that dependent claim 14 is allowable at least because of its dependence from claim 13 for the reasons discussed above.

Examiner respectfully disagrees. Dependent claim 14 is unpatentable over Wookey for the reasons discussed above with respect to claim 13.

Art Unit: 2162

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

HUNG Q PHAM
Examiner
Art Unit 2162

October 19, 2005


Conferees

JOHN BREENE, Supervisory Patent Examiner, AU 2162.

SAFET METJAHIC, Supervisory Patent Examiner, AU 2161.

HEWLETT PACKARD COMPANY
INTELLECTUAL PROPERTY ADMINISTRATION
3404 E HARMONY ROAD
P.O. BOX 272400
FORT COLLINS, CO 80528-9599


SAFET METJAHIC
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100


JOHN BREENE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100